

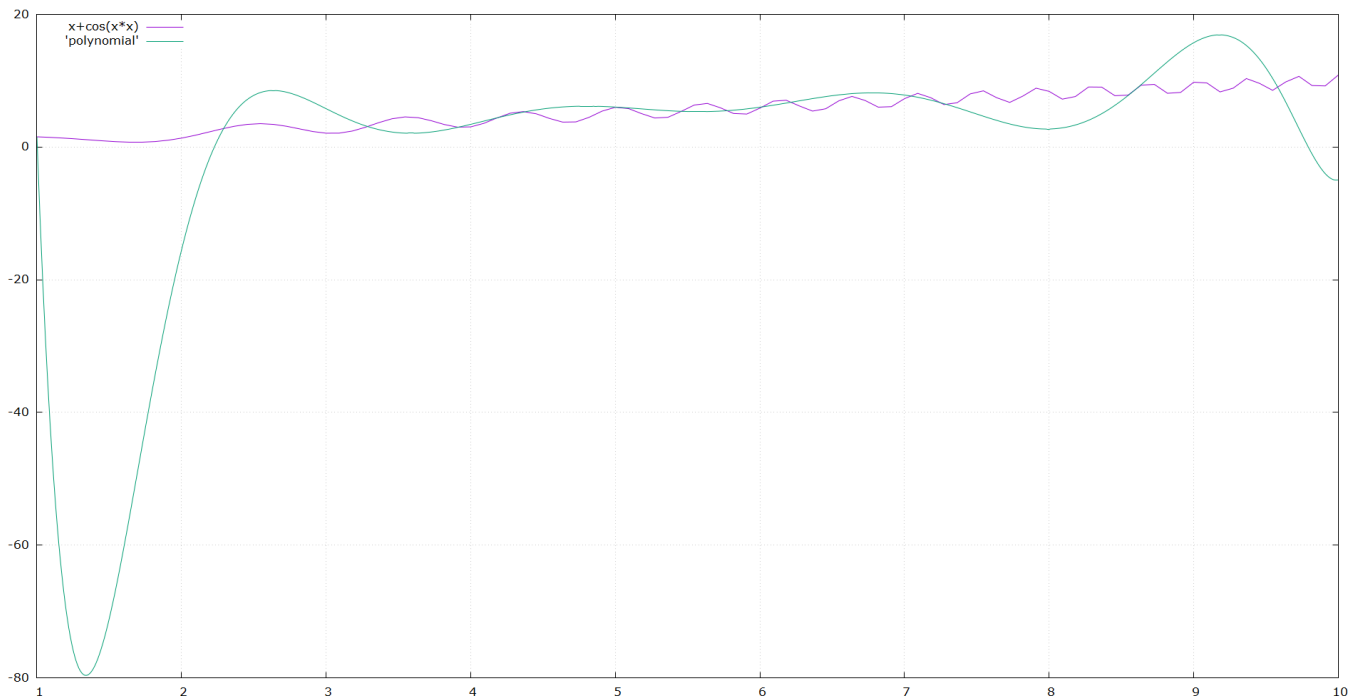
Metody obliczeniowe w nauce i technice - sprawozdanie 6

Łukasz Jezapkowicz

26.04.2020

- 1 Proszę skompilować i uruchomić program *interpolacja.c*. Korzystając z programu gnuplot narysować wykres. Narysować na jednym wykresie krzywe otrzymane różnymi metodami interpolacji (w przykładzie ustawione jest *gsl_interp_polynomial*).

Na początku wykres interpolacji poprzez wielomiany (prosta komenda `plot x+cos(x*x), 'polynomial' wl`).



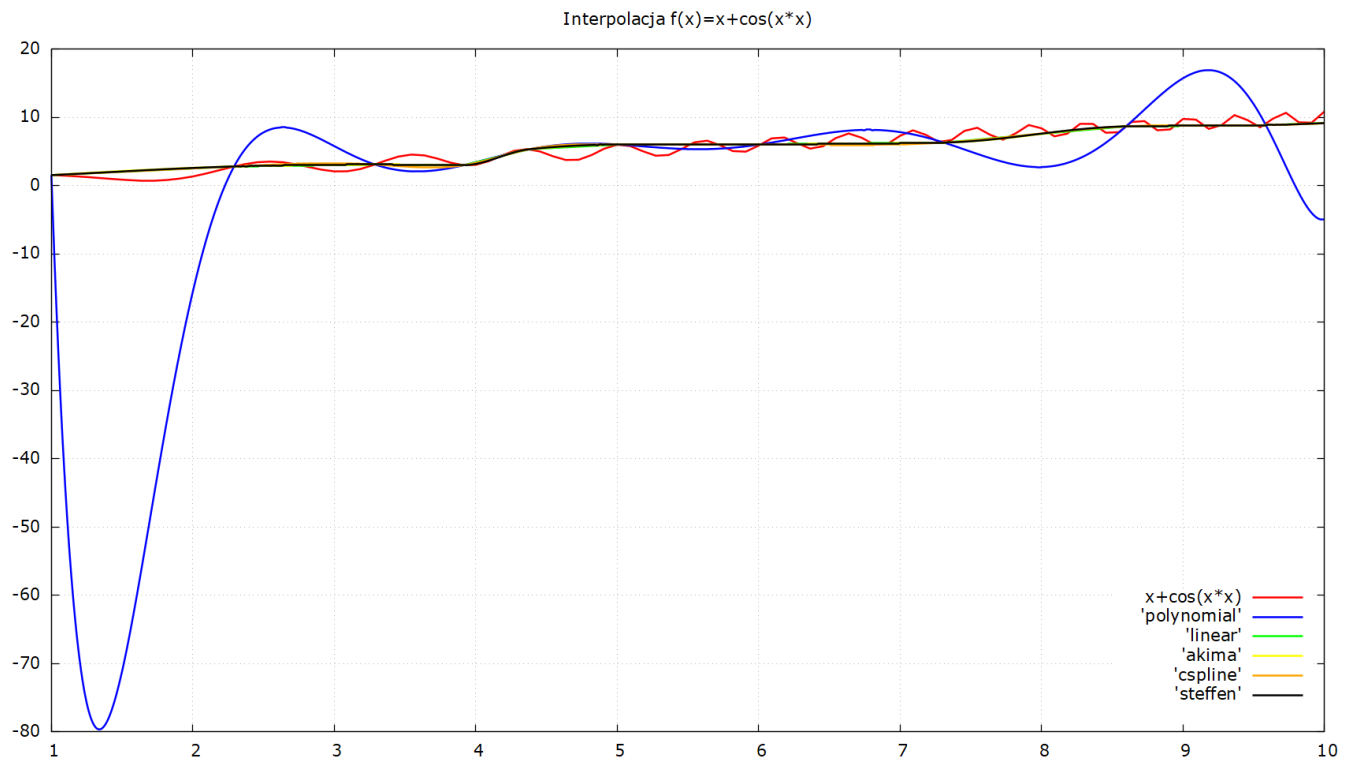
Poniżej widoczna zawartość pliku *zad1.gnu*, który pozwala stworzyć poprawne wykresy dla różnych typów interpolacji (*polynomial*, *linear*, *akima*, *cspline*, *stef fen*).

```
# program rysujący wykres interpolacji dla różnych sposobów interpolacji

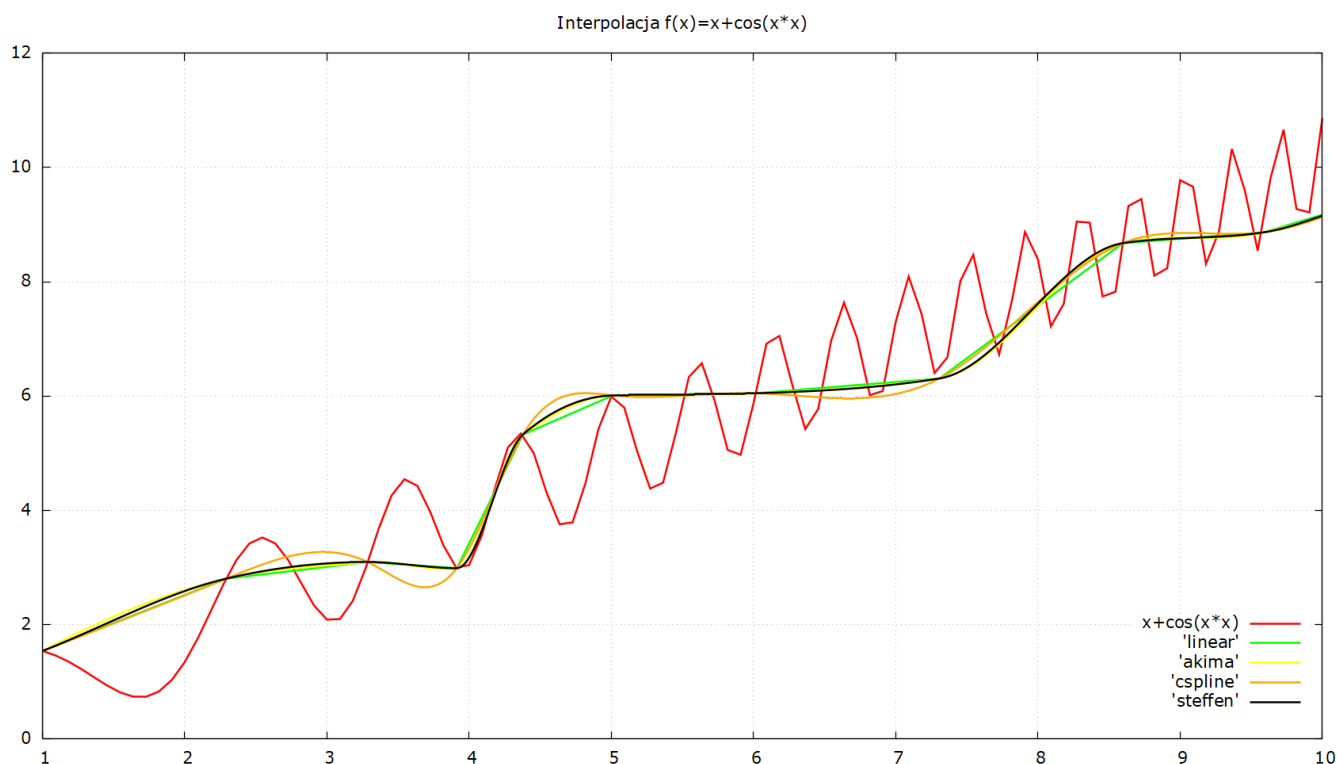
set term wxt size 1400, 800 enhanced font 'Verdana,10' persist
set size 1,1
set grid
set title "Interpolacja  $f(x)=x+\cos(x*x)$ "
set autoscale
set style line 1 lc rgb 'blue' lw 2
set style line 2 lc rgb 'green' lw 2
set style line 3 lc rgb 'red' lw 2
set style line 4 lc rgb 'yellow' lw 2
set style line 5 lc rgb 'orange' lw 2
set style line 6 lc rgb 'black' lw 2
set key right bottom

plot x+cos(x*x) ls 3, \
    'wartosci' ls 3 w p, \
    'linear' ls 2 w l, \
    'akima' ls 4 w l, \
    'cspline' ls 5 w l, \
    'steffen' ls 6 w l, \
```

Poniżej widoczny wynik działania komendy "load zad1.gnu".



Jak widać w przypadku interpolacji wielomianami wystąpił efekt Rungego. Poniżej wykresy wykluczając interpolacje wielomianami.



Wniosek: w programie GNUPLOT bardzo łatwo rysować wykresy na podstawie danych stworzonych przez bibliotekę *GSL*. W naszym przypadku żadna interpolacja nie oddała poprawnie wyglądu funkcji. Wynikło to z małej liczby węzłów, w których interpolowaliśmy.

2 Przy pomocy gnuplot proszę narysować dane zgromadzone w pliku *dane1.dat*. Aby wykres był czytelny, jedna z osi musi mieć skalę logarytmiczną. Proszę ustalić, która to oś i narysować wykres.

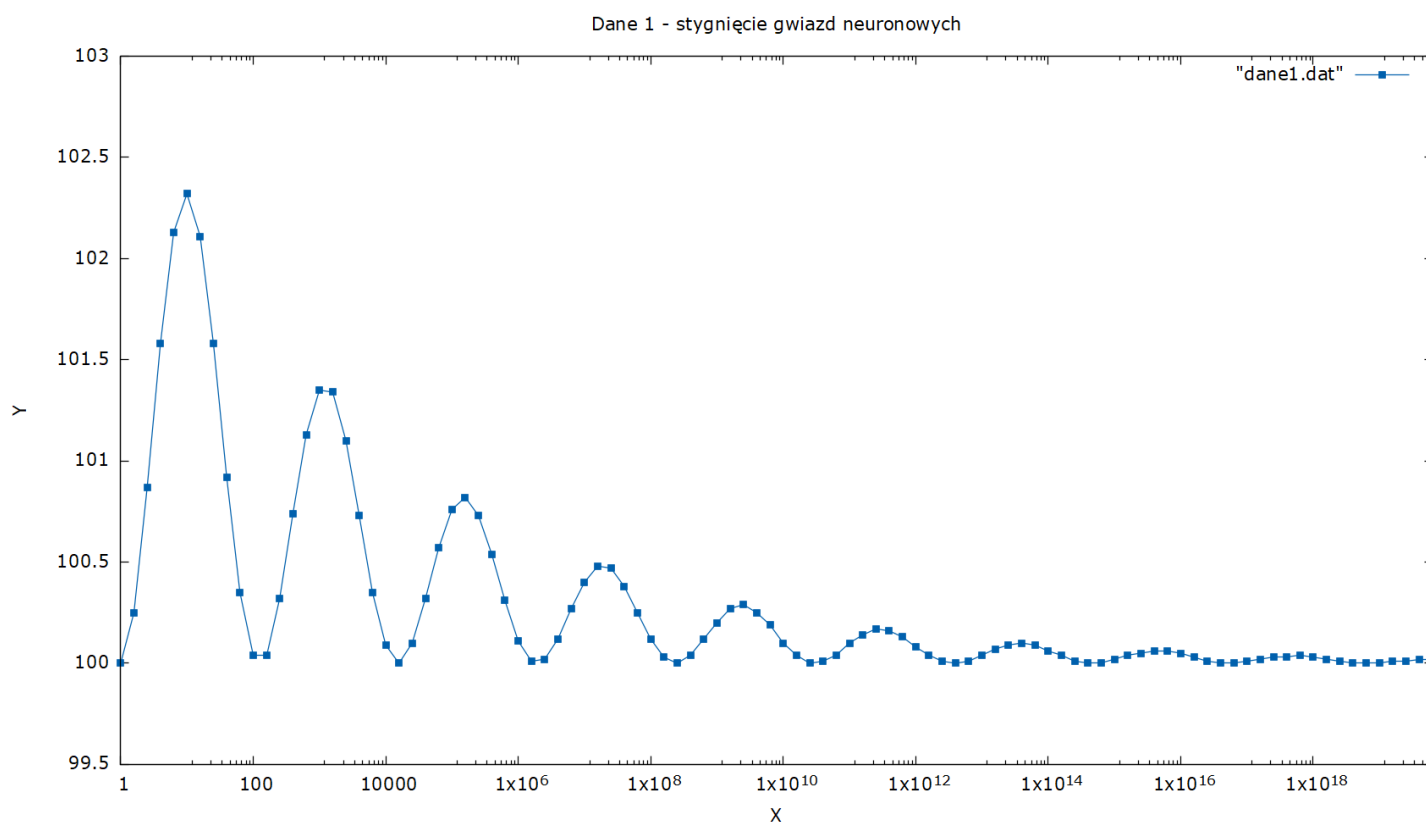
Przyglądając się danym, można zauważyć, że wartości x mieszczą się w przedziale $[1, 6.31 \times 10^{19})$ zaś wartości y mieszczą się w przedziale $[100, 103)$. Łatwo więc stwierdzić, że skalę logarytmiczną powinniśmy przyjąć na osi OX . Poniżej widoczna zawartość pliku *dane1.gnu*, który pozwala stworzyć poprawny wykres dla danych z pliku *dane1.dat*:

```
# program rysujący wykres dla danych z pliku dane1.dat

set term wxt size 1400, 800 enhanced font 'Verdana,10' persist
set size 1,1
set logscale x 10
set title "Dane 1 - stygnięcie gwiazd neuronowych"
set xrange [1:6.31e+19]
set yrange [99.5:103]
set xlabel "X"
set ylabel "Y"
set style line 1 lc rgb '#0060ad' pt 5

plot "dane1.dat" w lp ls 1
```

Poniżej widoczny wynik działania komendy "load *dane1.gnu*".



Wniosek: w programie GNUPLOT bardzo łatwo odpowiednio skalować wykresy np. do skali logarytmicznej. W programie powyżej zastosowałem logarytm o podstawie 10.

3 Proszę narysować wykres funkcji dwuwymiarowej, której punkty znajdują się w pliku *dane2.dat*. Proszę przeglądnąć plik i spróbować znaleźć w nim maksimum. Potem proszę zlokalizować maksimum wizualnie na wykresie. Proszę na wykresie zaznaczyć maksimum strzałką

Poniżej widoczna zawartość pliku *dane2.gnu*, który pozwala stworzyć poprawny wykres dla danych z pliku *dane2.dat*:

```
# program rysujący wykres dla danych z pliku dane2.dat

set term wxt size 1400, 800 enhanced font 'Verdana,10' persist
set size 1,1
set grid
set title "Dane 1 - wykres w 3D"
set autoscale
set xlabel "Oś OX"
set ylabel "Oś OY"
set zlabel "Oś OZ"
set arrow 1 from 4,3,1.5 to 4,3,1
set style line 1 lc rgb '#0060ad' pt 5

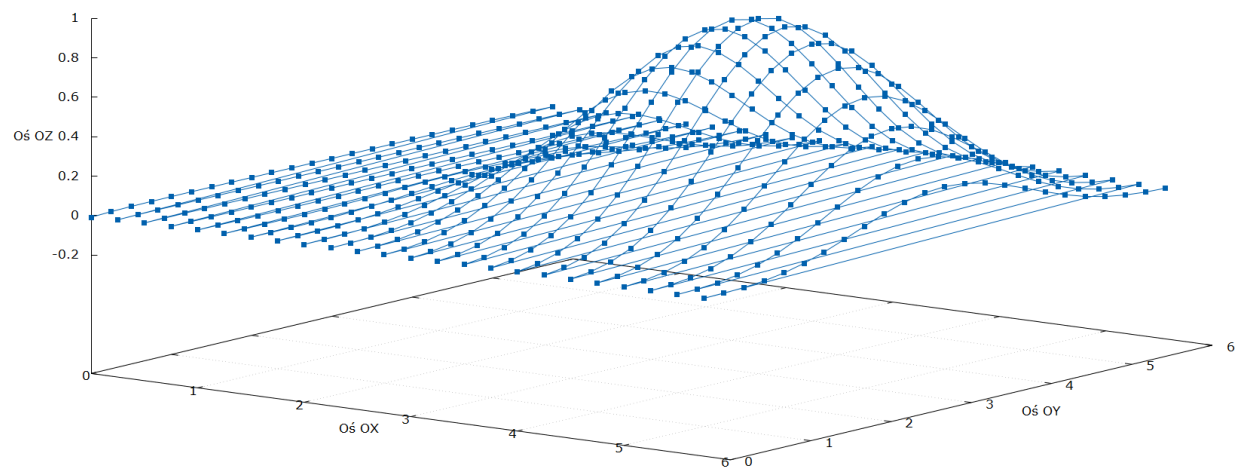
splot "dane2.dat" w lp ls 1
```

Przeglądając plik *dane2.dat* można zobaczyć, że swoje maksimum lokalne funkcja przyjmuje w punkcie (4,3).

Poniżej widoczny wynik działania komendy "load *dane2.gnu*" pod różnymi kątami.

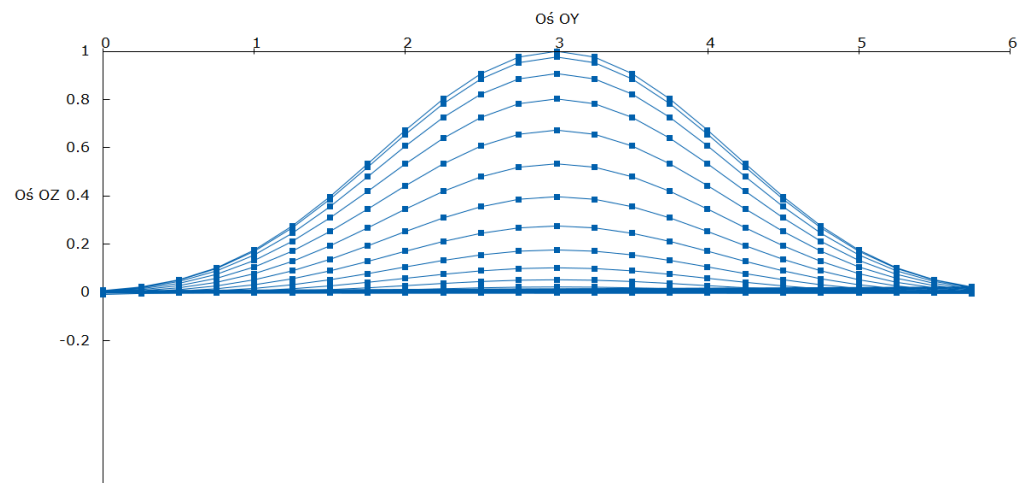
Dane 1 - wykres w 3D

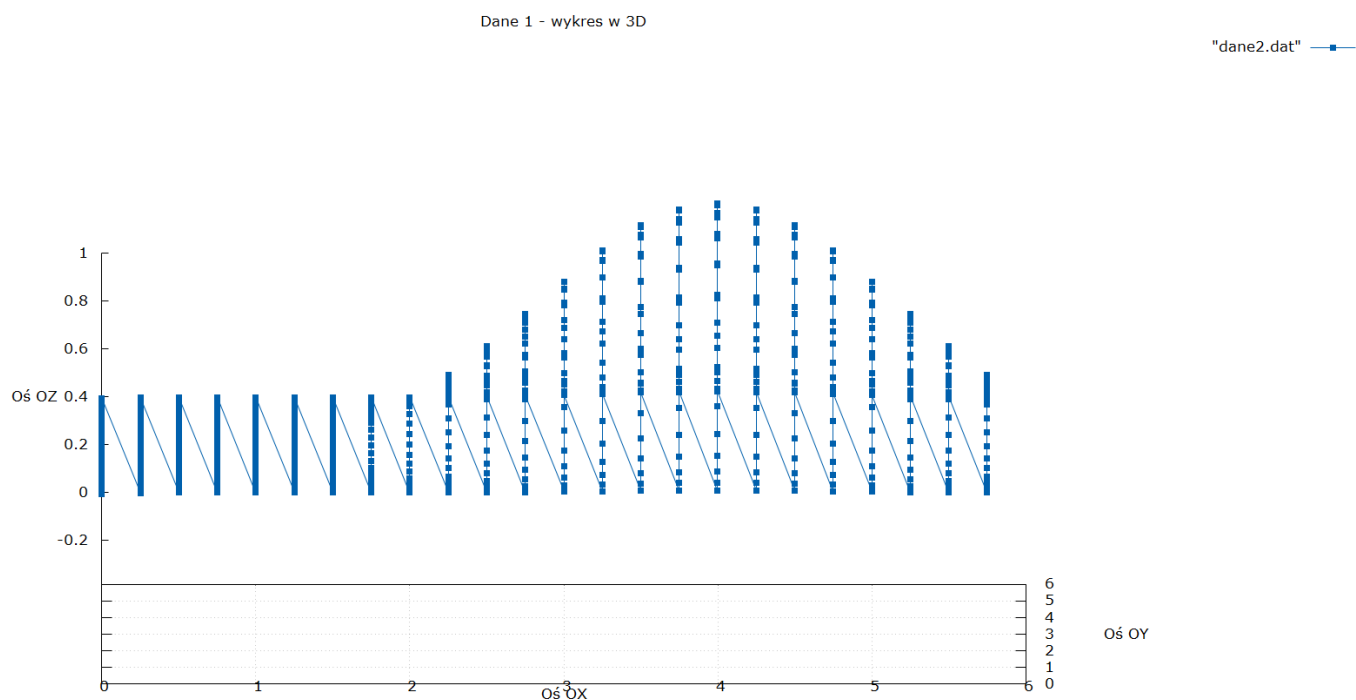
"dane2.dat" —



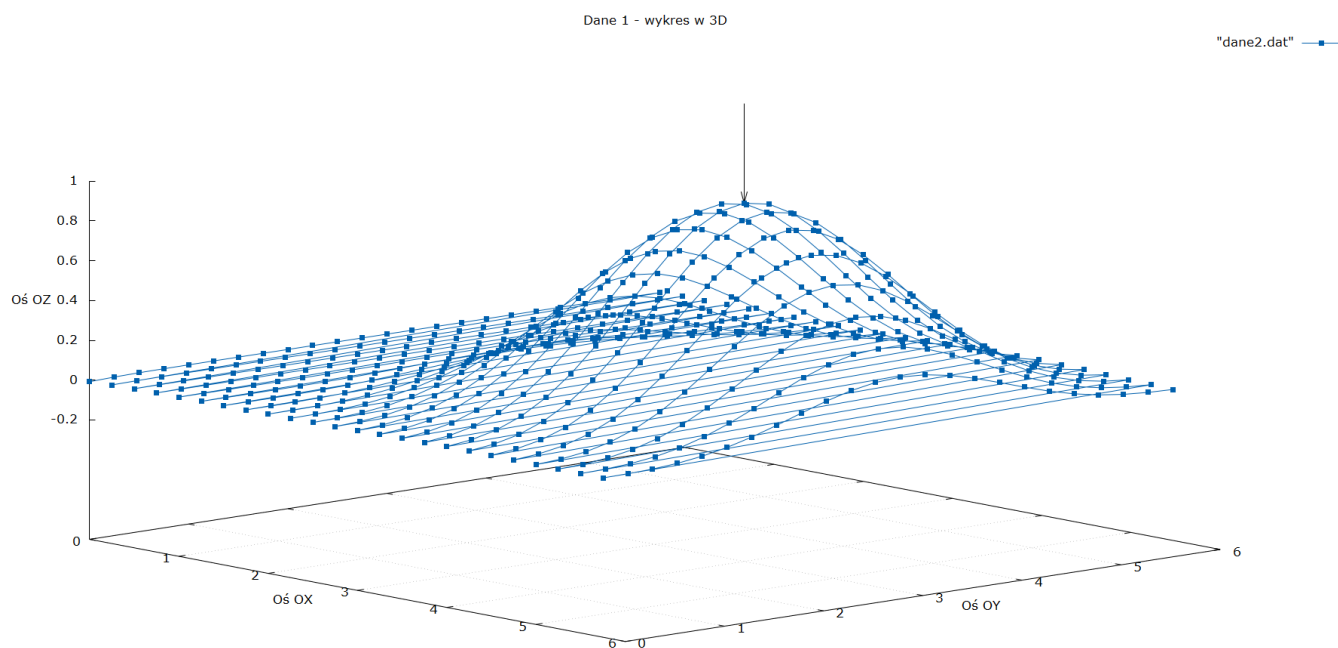
Dane 1 - wykres w 3D

"dane2.dat" —



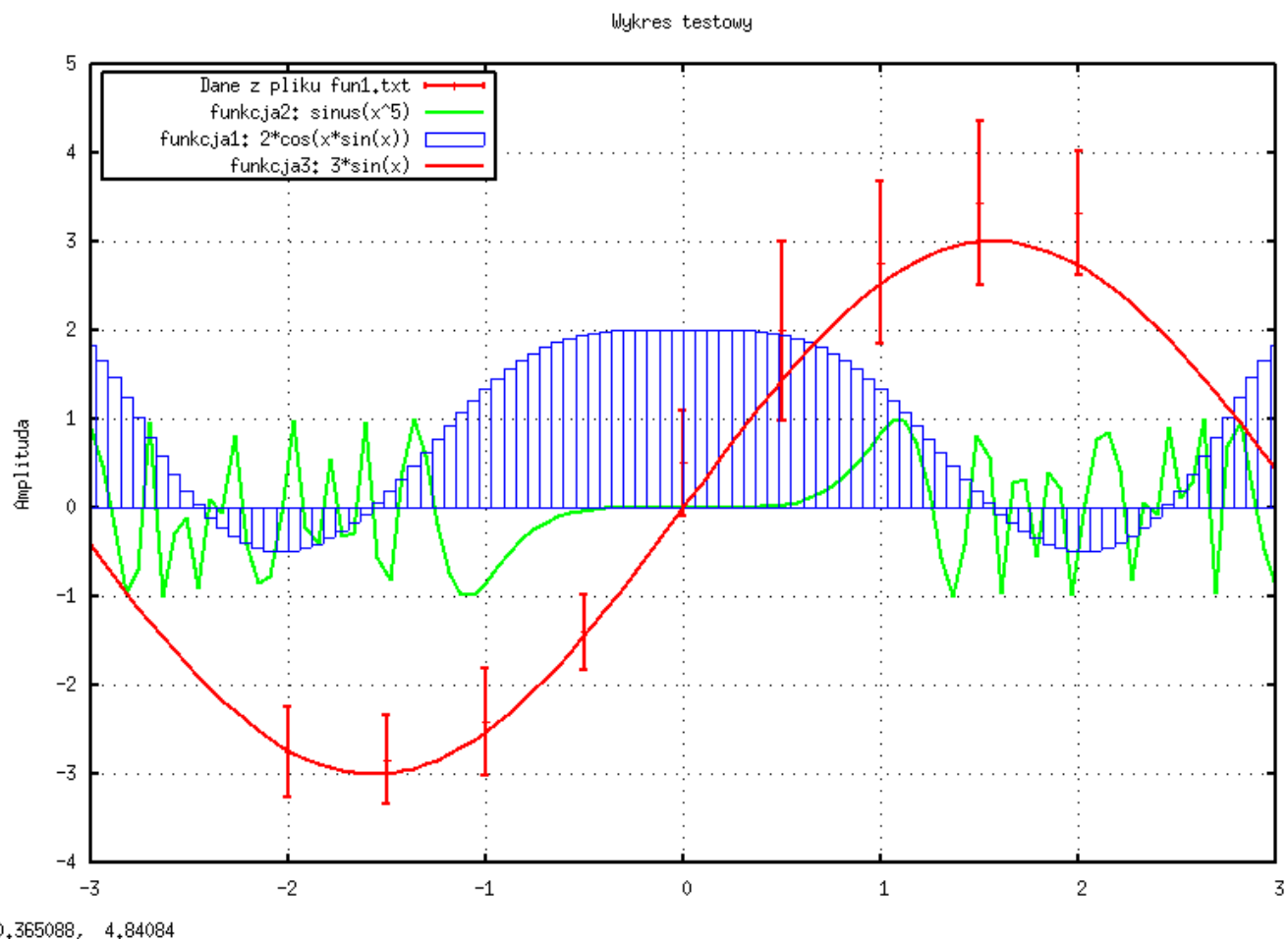


Z powyższych wykresów łatwo zauważyć, że maksimum faktycznie osiągnięte jest w punkcie (4,3).
Poniżej widoczny również wykres z zaznaczonym maksimum.



Wniosek: w programie GNUPLOT bardzo łatwo rysować wykresy w przestrzeni trójwymiarowej. Możliwość rotacji wykresem pozwala nam bardzo łatwo znaleźć interesujące informacje na temat funkcji np. jej maksimum.

4 Proszę odtworzyć wykres znajdujący się na rysunku:



Pierwszą rzeczą, którą należy zrobić to odczytać przybliżone wartości punktów z pliku *fun1.txt* wraz z ich niepewnościami. Odczytane i oszacowane przeze mnie wartości widać poniżej:

-2	-2.6	0.5
-1.5	-2.7	0.5
-1	-2.4	0.6
-0.5	-1.4	0.4
0	0.5	0.55
0.5	2	1
1	2.7	0.9
1.5	3.4	0.9
2	3.3	0.7

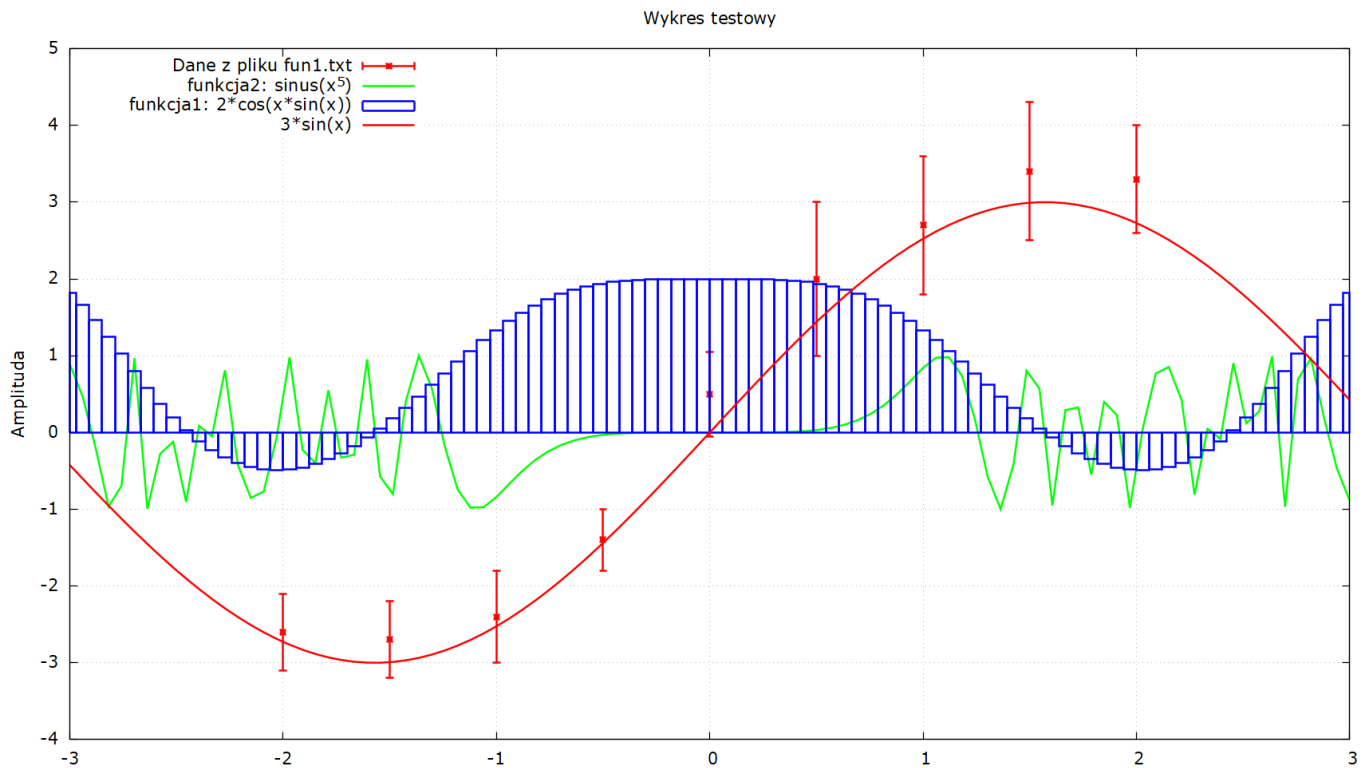
Poniżej widoczna zawartość pliku *dane3.gnu*, który pozwala odtworzyć wykres z rysunku:

```
# program odtwarzający wykres z zadania 3

set term wxt size 1400, 800 enhanced font 'Verdana,10' persist
set size 1,1
set grid
set title "Wykres testowy"
set xrange [-3:3]
set yrange [-4:5]
set ylabel "Amplituda"
set style line 1 lc rgb 'blue' lw 2
set style line 2 lc rgb 'green' lw 2
set style line 3 lc rgb 'red' lw 2
set key left top

plot 'fun1.dat' w errorbars ls 3 title 'Dane z pliku fun1.txt', \
    sin(x**5) w l ls 2 title 'funkcja2: sinus(x^5)', \
    2*cos(x*sin(x)) w boxes ls 1 title 'funkcja1: 2*cos(x*sin(x))', \
    3*sin(x) with l ls 3 'funkcja3: 3*sin(x)'
```

Poniżej widoczny wynik działania komendy "load *dane3.gnu*":



Wniosek: w programie GNUPLOT bardzo łatwo odpowiednio odwzorowywać odpowiednie wykresy oraz manipulować ich wyglądem (kolor, typ linii, wielkość).